UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/673,587	09/29/2003	Jonathan Appavoo	YOR920030317US1 (16856)	1607

23389 7590 12/04/2007
SCULLY SCOTT MURPHY & PRESSER, PC
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
12/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

HK

Office Action Summary	Application No.	Applicant(s)	
	10/673,587	APPAVOO ET AL.	
	Examiner	Art Unit	
	Tuan A. Vu	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 10/12/07.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-24 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-24 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.	6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 10/12/07.

As indicated in Applicant's response, claims 1, 4-10, 12-24 have been amended. Claims 1-24 are pending in the office action.

Claim Objections and compliance to CFR § 1.121

2. Claim 21 is objected to because of the following informalities: the claim includes some amendments but is marked as "(Original)". This is not compliant with CFR § 1.121c for which a non-compliant response type of infraction would be in order. This informality is however treated herein as a minor informality in order to expedite prosecution of the case.

3. Claims 1, 10 and 18 are objected because the 'source code component' phrase therein has been replaced with 'code component' and no proper marking is in place to indicate that the term 'source' has been deleted. Claim 10 for reciting 'for swapping source code in a computer system' appears to present a typographical inconsistency because nowhere in the body of the claim is there any trace of 'source code'.

4. Likewise, claims 4, 6, 9, 10, 12-14, 16-17, 19, 21, 24 recite newly added limitations without having proper markings to indicate deleted text (e.g. *mechanism is divided across, so that each processor can proceed, whereby, quiescent state*); and like above, the amendments as presented are not compliant with § 1.121c. This repetitive type of informality is however treated herein as a minor informality in order to expedite prosecution of the case.

5. Claims 7 recite added limitations without proper underlying of text (e.g. references, li. 5) and without proper markings to indicate deleted text (e.g. quiescent state, li. 8) in accordance to

CFR § 1.121; and claim 23 recites 'transferring the *identified at* said quiescent state' (li. 8) and this is a treated as a typographic impropriety.

In all, non-compliancy to CFR § 1.121 is a impropriety that would trigger a Office Action noticing about non-compliant Amendment under that rule and would obviate a full examination of the merits; but in order to prosecute the case, the above would be treated as mere claims objections, all of which whenever possible require corrective action.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 6, 9, 14, 17, 21, 24 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 6, 9, 14, 17, 21, 24 recite 'and *generating to* identify said references to said objects, which are entered in the table' (line 4 of respective claims). It is not clear what entity has been generated for the purpose of identifying references. This lack of definiteness will be treated as identifying 'references to' said objects such that the references are entered in the table.

8. Claim 10-17 are rejected for indefinite language in that claim 17 recites the limitation "transferring the identified references to the *second code* component" in line 8, when 'second code component' has no proper reference earlier in the claim. There is insufficient antecedent basis for this limitation in the claim. Claims 11-17 for failing to remedy to this deficiency are also rejected; and this second component will be treated as a component replacing the first component.

Correction is required.

Double Patenting

9. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

10. Claims 6, 14, 21 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 18 of copending Application No. 11/227,761 (hereinafter '761).

Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observations. Following are but a few examples as to how the certain claims from the instant invention and from the above copending application are conflicting with each other.

As per instant claim 6, '761 claim 1 also recites dynamically update an operating system (Note: this is equivalent to *while said operating system remains active ... provides continual availability of hardware resources by applications in the computer system* of claim 6) comprising identifying references (i.e. *reference pointer*) to said first code component and replacing the identified references to said first code component to said new code component (i.e. *changing a factory reference pointer ... to the new factory object*). '761 claim 1 does not explicitly recite by separating the first code component into objects grouped in table, whereby identified references to said objects are entered in the table; but in view of the suggestion by '761 that the reference pointer is changed to point to factory object to new factory object (during the *loading of a new factory object*) and removing the old factory object, it would have been obvious for one skilled in the art to provide said runtime factory object in form of runtime table entry as in a table storing reference entries (each being a pointer); so that these table-represented reference pointers would be identifying during the dynamic replacement of reference objects as recited above, in view of well-known approach using reference table to interrelate dynamic code referencing to provide program references resolution support at runtime.

As per instant claims 14, and 21, these claims recite the main limitations of instant claim 1, hence would also have been obvious variations of '761 claim 1 in light of the above analysis.

As per claims 6, 14, and 21, '761 claim 18 also recites updating of an operating system without rebooting, identifying references (i.e. *determining ... old class definition meets ... requirement of a new class definition; structures indicate ... instantiations of the old class definition* -- Note: structure representing definition of old or new class reads on references of first or new code component; i.e. said structure being identified for said determining leading to the hot-swapping) to said first code component and replacing the identified references to said first code component to said new code component (i.e. *hot swapping each ... object instance for its corresponding old object instance*). '761 claim 18 does not explicitly recite by 'separating the first code component into objects grouped in table, whereby references to said objects are entered in the table'; but in view of the structure to store the definition referring to the *object* *instance* being swapped from above, this reference *table* (groups of object reference) limitation would have been obvious in light of the rationale as set forth above.

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

12. Claims 1-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Slingwine et al., USPN: 6,219,690 (hereinafter Slingwine)

As per claim 1, Slingwine discloses in a computer system using an operating system to provide access to hardware resources (e.g. data coherence … hardware requirement … cache memories -col. 5, lines 4-20; shared-memory - col. 6, lines 41-61), wherein said operating system provides access to said resources via a first source code component, a method of replacing said first source code component with a new source code component while said operating system remains active and while said operating system provides continual availability to applications of the hardware resources (see Fig. 2; *kernel running* - col. 10, li. 54-67) by applications operational in the computer system, the method comprising:

identifying references to said first code component (e.g. current generation 108 – pg. 3 – Note: CALLBACK processing to make next generation to become current generation – see col. 6, lines 41-50 – whereby associated structured context information for a thread activity – --- col. 9, li. 45-to col 10, li. 17 – along with generation-contained callback constructs – see Fig. 4;

handle table, *table_ptrs*, Fig. 6-- are replaced **read on** identifying references of first code component); and

replacing the identified references to said first source code with references to said new source code component (next generation 110, Fig. 3; UPDATES: Add to next generation 90 – Fig. 3; Fig. 4).

As per claims 2-3, Slingwine discloses wherein the method is implemented transparently to said applications (e.g. Fig. 5A, 5B, 5C, 5D; *interrupt* -- col. 18, lines 42-55 – Note: interrupt periodically invoked without need for user input reads on mutual exclusion of threads via callback implementation being transparent to user – see *kernel 36, interrupt* under *user process* – Fig. 2); wherein the method is scalable (e.g. *global generation, possible large number of processes* – see col. 18, lines 42-46; *expanded* – col. 17, lines 41-67; *per-thread where possible ... some other entity where possible* – col. 10, li. 44-47).

As per claim 4, Slingwine discloses a multiprocessor system with plurality of processors so that the method is implemented on each processor independently (e.g. *per-processor context* - Fig. 4; col. 19, li. 1-3).

As per claim 5, Slingwine discloses

establishing a quiescent state for the first code component (e.g. col. 7, lines 50-67);

transferring said identified references at said established quiescent state (e.g. *unlinking ... adds a callback to the next generation... links the new element ... erases the original element* – col. 8 lines 54-67; Fig. 3; col. 9, lines 18-44) from the first code component to the new code component; and after transferring said identified references to the new code segment at said established quiescent state (e.g. *through a quiescent state ... allowing updater ... safely* - col. 10, lines 7-51; col. 8, lines 54-67), swapping the first code component with the new code component (e.g. col. 7, lines 50-67; col. 10, lines 7-51).

As per claim 6, Slingwine discloses separating the first code component into objects; and grouping said objects into a table (e.g. Fig. 6; *one bit per thread ... next level ... group of threads and so on* – col. 9, lines 31-44; col. 12, lines 5-15 – Note: thread and associated data structures per threat activity reads on objects and references thereto being grouped into structure – see col. 12, lines 35-53), whereby references to said objects are entered in the table (e.g. col. 12, lines 35-53; *handle table*, *table_ptrs*, Fig. 6).

As per claim 7, Slingwine discloses wherein the replacing step includes the steps of: establishing a quiescent state for the first code component, without locking the first code component, by tracking active threads to the first code component and identifying active threads as said references (refer to claim 5); transferring said identified references during the quiescent state from the first code component to the new code component; and after transferring said said identified references, swapping the first code component with the new code component (refer to claim 5).

As per claim 8, Slingwine discloses wherein the replacing step includes the steps of: establishing a quiescent state for the first code component that includes the identified references (refer to claim 7); transferring the identified references at the quiescent state from the first code component to the new code component by providing a infrastructure operating a best transfer algorithm (see Fig. 3-5 – Note: mutual exclusion policy via passing of changes as context structure updates to the next thread context reads on best algorithm infrastructure); and after transferring said quiescent state, swapping the first code component with the new code component (refer to claim 7); and further discloses transferring by providing an infrastructure to negotiate a best transfer algorithm (refer to claim 7).

As per claim 9, Slingwine discloses wherein the step of identifying references includes the steps of

separating the first code component into objects, and grouping said objects into a table, whereby identified references to said objects are entered in the table (refer to claim 6); and

the replacing step includes the steps of

establishing a quiescent state for the first code component that includes the identified references; transferring the the identified references at the quiescent state from the first code component to the new code component by providing an infrastructure to negotiate a best transfer algorithm; and after transferring said references to the new code, swapping the first code component with the new code component (refer to claims 5, 7, or 8).

As per claim 10, Slingwine discloses a system for swapping source code in a computer system including an operating system, said operating system including at least one source code component and providing continual availability to applications of hardware resources (refer to claim 1) by applications operational in the computer system, the system comprising:

means for identifying, while said operating system is active and providing continual access to said resources, references to a first source code component of the operating system; and

means for replacing the identified references, while said operating system is active and providing continual access to said resources (*interrrupt* -- col. 18, lines 42-55 ; *kernel 36,* *interrupt* under *user process* – Fig. 2), to said first source code with references to a new source code component for the operating system;

all of which steps of identifying and replacing having been addressed in claim 1.

As per claim 11, refer to the rejection of claim 2-3.

As per claims 12-17, refer to claims 4-9, respectively.

As per claim 18, Slingwine discloses a program storage device, for use with a computer system including an operating system to provide access to hardware resources, wherein said operating system provides access to said resources via a first source code component (Fig. 1-2), said program storage device being readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for replacing said first source code component with a new source code component while said operating system remains active and while said operating system provides continual availability to applications of said resources by applications operational in the computer system(refer to claim 1), the method steps comprising:

identifying references to said first source code component; and

replacing the identified references to said first source code with references to said new source code component;

all of which being addressed in claim 1.

As per claims 19-24, refer to claims 4-9, respectively.

Response to Arguments

13. Applicant's arguments filed 10/12/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

Double-Patenting rejection:

(A) Applicants have submitted that because the subject matter of independent claims 3, 13, and 1 from which claims 6, 14, 21 are derived is patentably distinct from claims 1, 18 of '761 application (Appl. Rmrks pg. 11, middle). Claims 6, 14, and 21 subject matter is dependent from claims 1, 10, and 18 respectively, and the rejection has pointed out analogous teachings from reading and interpreting the language of the claims 6, 14, 21 when compared with the teaching derived from '761 claims 1, and 18. The rationale for obviousness has been based on broad interpreting and knowledge of one of ordinary skill in the art when some teaching from the above '761 subject matter suggests some motivation to combine or to render obvious what appears to be just a mere syntactic or language usage differential between the conflicting subject matter of the claims being compared. Applicants fail to point out why such difference in language is insufficient to yield grounds for the DP rejection; i.e. one skill in the art to construe an obvious variation of language for conveying analogous inventions based on the teachings understood parsing the pertinent language from '761. Hence, Applicants' argument amount to mere allegation of distinct subject matter being submitted without sufficient grounds as to overcome the rejection.

35 USC § 102 Rejection:

(B) Applicants have submitted that the hot-swapping by the invention is seamless to hardware resources based on quiescent state transfer of references from the first code component to a new code component; and that distinguishes with Slingwine's mutual-exclusion, wherein updating of data to maintain coherency is based on data-updating of threads via pass through of a quiescent state from attempt to update data to a time data are actually updated (Appl. Rmrks pg.

14); i.e. Slingwine does not swap a first code component wit a new code component as in claims 1, 10, and 18. Claims 1, 10 and 18 do not recite replacing of a first code component with a new code component by transferring references to the new code component. Transferring references related to the first component onto another component that take over from the first component amounts to switching context from a current thread executing with a newer context executing under another thread. The rejection is not specifically addressing a hot-swapping as alleged above but content with fulfilling replacing one thread with another by transferring references from said first thread to a context in which another thread would be supplanting the first thread; i.e. replacing the identified references to said first code component with references to said new code component. The claim does not elaborate particular teaching with regard to the nature of the 'new code component'; hence broad interpretation of 'new' as applied in the rejection revolves around the connotation that an previous thread context has been supplanted by a replacing thread context, the supplanting being equivalent to substituting a older context with a more recent one as in older versus newer. Further, the 'replacing' of code as entailed from the claim preamble is not establishing a alleged OS seamless hot-swapping (which is not claimed) but merely a replacement of one code component by another to maintain availability of resources in that hardware resource access would be without interruption; based on broad interpretation of claim 1, 10, and 18. The crux of the claimed subject matter is about replacing by transferring of references pertinent to the first component to the replacing component. Slingwine discloses a memory access (or hardware resource/ shared-memory access) using a proper algorithmic approach by which some thread would replace another thread execution based on some quiescent conditions being reached; and such replacement process would be equivalent to accessing hardware resources context in which a first component is replaced by another, based on transfer process by which updating of pertinent references or thread data (references respective to first or new component in regard to the replacement or switching of threads) in as presented Slingwine's structure updates set forth in the rejection. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

(C) Applicants have submitted that Slingwine's mutual exclusion applies quiescent state not at the very moment where the thread is accessing their exclusion mechanism (Appl. Rmrks, pg. 15, middle para). This analysis has no direct connection with the actual state of the claim language for no specific claim language has been identified when the above allegation is made. Applicants assert that Slingwine's mutual exclusion is not to describe providing access to hardware resources and 'replacing said first code component … while said operating system … active … continual availability … by applications operational in the computer system'. The claims as a whole have been interpreted and amount to hardware resource access uninterrupted by way of identifying references to a first code component, and replacing said first component by transferring those references to the new component that is replacing the first code component. The claims as questioned from the above (Note: the Applicants fail to specify any particular language when making the above allegation) in all, do not provide specificity in such details as to preclude Slingwine's exclusion method from reading onto the above teaching derived from broad interpretation of the claim. That is, when a thread is passed through quiescent state checking and deemed that it can be replaced by another thread, the shared memory access is maintained; and the exclusion method of updating structures (see Rejection) by Slingwine is deemed sufficient to fulfill transferring of references to the replacing code component. The argument is not persuasive because it does not point out what claim language the argument is concerned with; nor does it establish a clear teaching in a claim language that would convince that Slingwine's substitution distinguishes over what is clearly recited. The argument is also referred to section B from above.

(D) Applicants have submitted that the portions cited in the Office Action do not teach 'identifying references to the first code component' as required (Appl. Rmrks pg. 15, bottom). It is deemed that a structure establishing information regarding a thread such that this structure includes thread counter, its calling back processor as above mentioned; or rcc_control variables including pointer references (* *rcc_args*, * *rcc*_next; * next rc_callback_t -- see Slingwine: col. 12: typedef struct *rc_callback_t, rc_ctrlblk*) would be ample evidence of references to a first thread prior to the update needed to switch context to a new thread based on mutual-exclusion and quiescent state pass through by Slingwine. It is therefore appropriate to reasonably state that the 'references to the first code component' limitation has been fulfilled.

(E) Applicants have submitted that Slingwine does not teach 'replacing the identified references to the first … with references to said new code component' (Appl. Rmrks pg. 16, 2nd para). Slingwine updating of the callback structure has been analogized to be replacing step by which switching context from one thread to the new thread includes all the references which were there for the first context to be dynamically updated to reflect the context of the newer thread, as set forth in the updates by Slingwine (see text of Fig. 3-5). Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references.

In all, the claims stand rejected as set forth in the Office Action.

Conclusion

14. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

[illegible]

Tuan A Vu
Patent Examiner,
Art Unit 2193
November 30, 2007